CPSC 440: Machine Learning

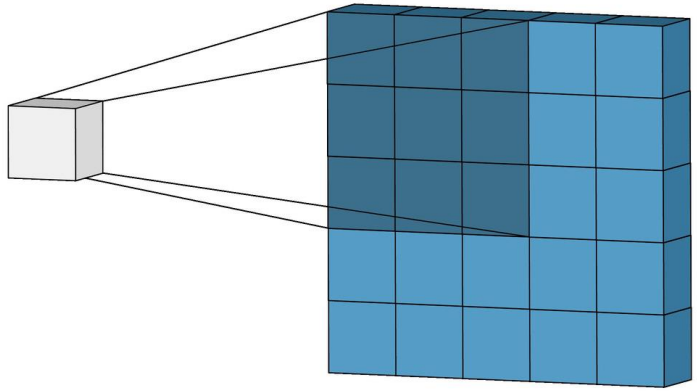
Convolutional Neural Networks

Winter 2022

Last Time: Convolutions

- We started to discuss convolutions:
  - Generate new image by applying filter.

$$z[i_1, i_2] = \sum_{j_1=-m}^{m} \sum_{j_2=-m}^{m} w[j_1, j_2] x[i_1 + j_1, i_2 + j_2]$$



- This was motivated by image classification problems:

"Abnormality detected"
(binary classification)

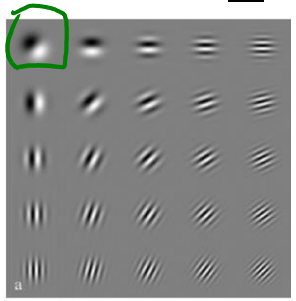
  - Applying several convolutions gives features that can help classification.

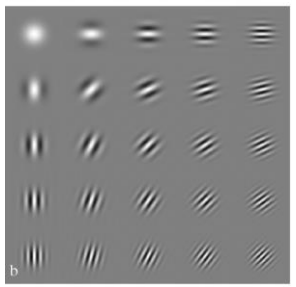
https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

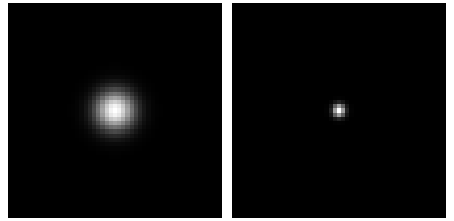
Convolutions

"signed" image:
- gray means 0
- white means 1
- dark means -1

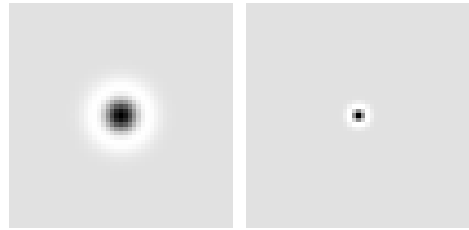
- Pre-2012, people often designed the filters by hand.
  - Filters can approximate "derivatives" or "integrals" of the image regions.
    - Derivative filters will up to 0, integral filters will add up to 1.
  - Three of the most-common filters that people used:







- Gaussian filters: integral filter, giving the average brightness in a region.
  - Variance of the Gaussian controls the amount of smoothness.
  - This produces a pixel feature that is less sensitive to noise than pixel's raw value.
- Gabor filters: derivative filters, measuring changes in brightness along a direction.
  - We typically compute these for different orientations and "frequencies".
  - This gives a set of features that is useful in describing edges in the image.

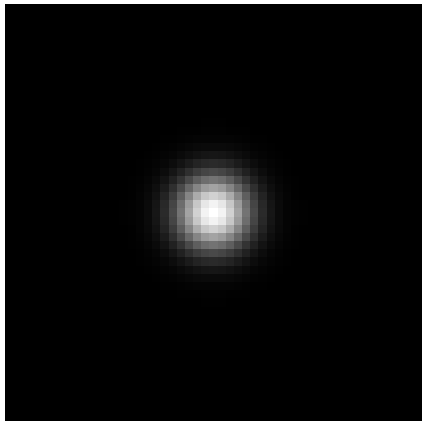


- Laplacian of Gaussian filter: total second-derivative filter.
  - Complements Gabor filters: helps describe if change is due to an edge, line, or continuous change.
  - Similar filters may be used early in the eyes visual processing.
  - I think of the results of convolutions as the "bag of words" making up images.

Image Convolution Examples

Gaussian Convolution:

$*$



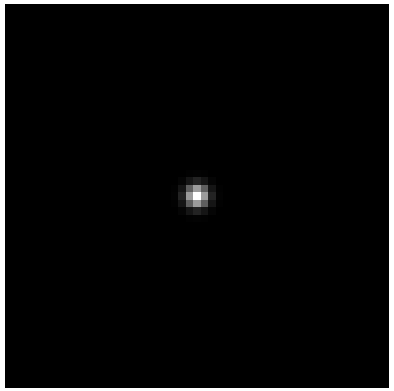
$=$

blurs image to represent average (smoothing)

Image Convolution Examples

Gaussian Convolution:

$*$



(smaller variance)

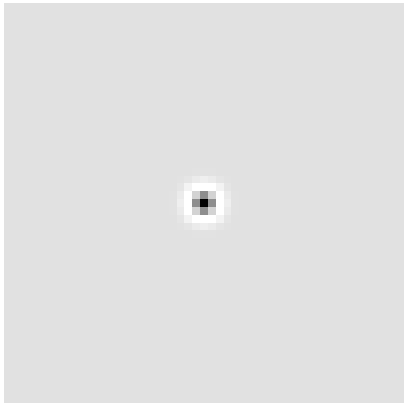
$=$

blurs image to represent average (smoothing)

Image Convolution Examples

Laplacian of Gaussian

*



=

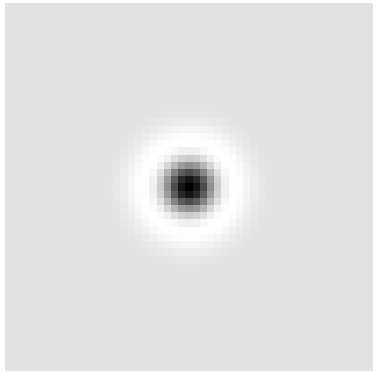
"How much does it look like a black dot surrounded by white?"

"signed" image (gray is 0)

Image Convolution Examples

Laplacian of Gaussian

$*$



(larger variance)

$=$

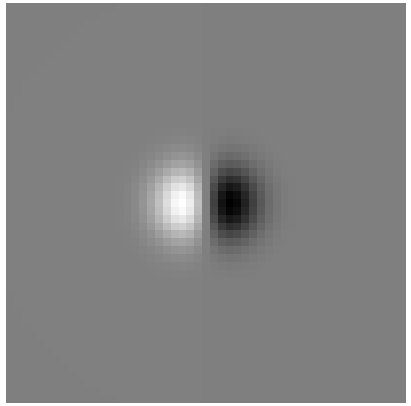
Black/white as sides of edge

Similar preprocessing may be done in basal ganglia and LGN.

Image Convolution Examples

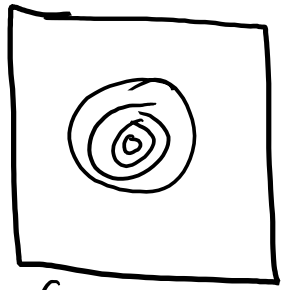
Gabor Filter
(Gaussian multiplied by sine or cosine)

$*$



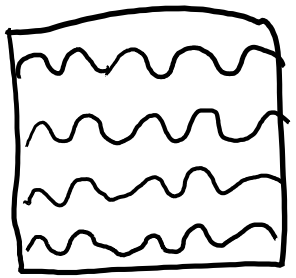
$=$

$=$



Gaussian

$*$



Parallel Sine functions

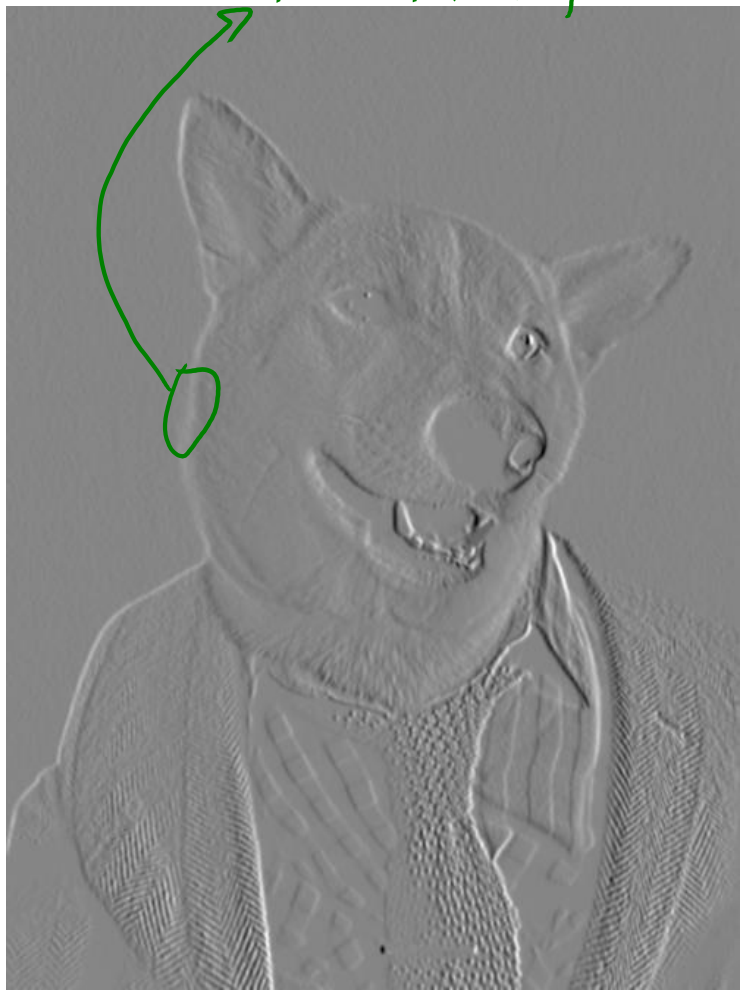
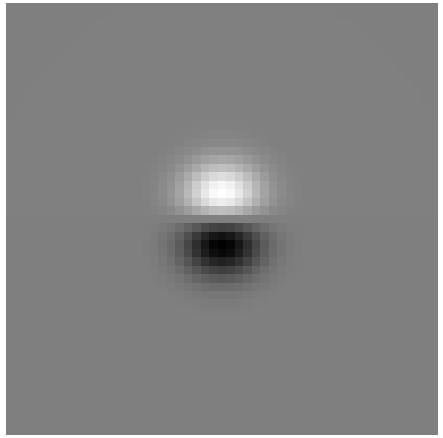
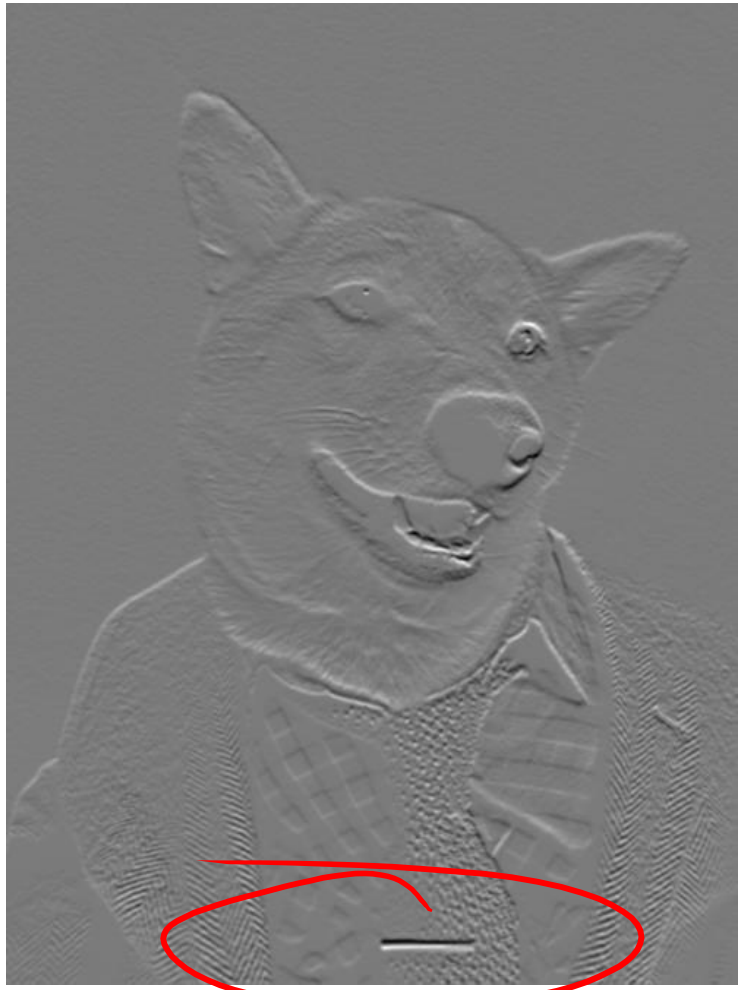


Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)



$*$ =

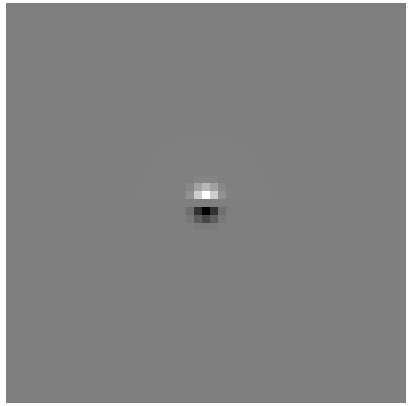


Different orientations of the sine/cosine let us detect changes with different orientations.

2d derivatives have a direction.

Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)



(smaller variance)

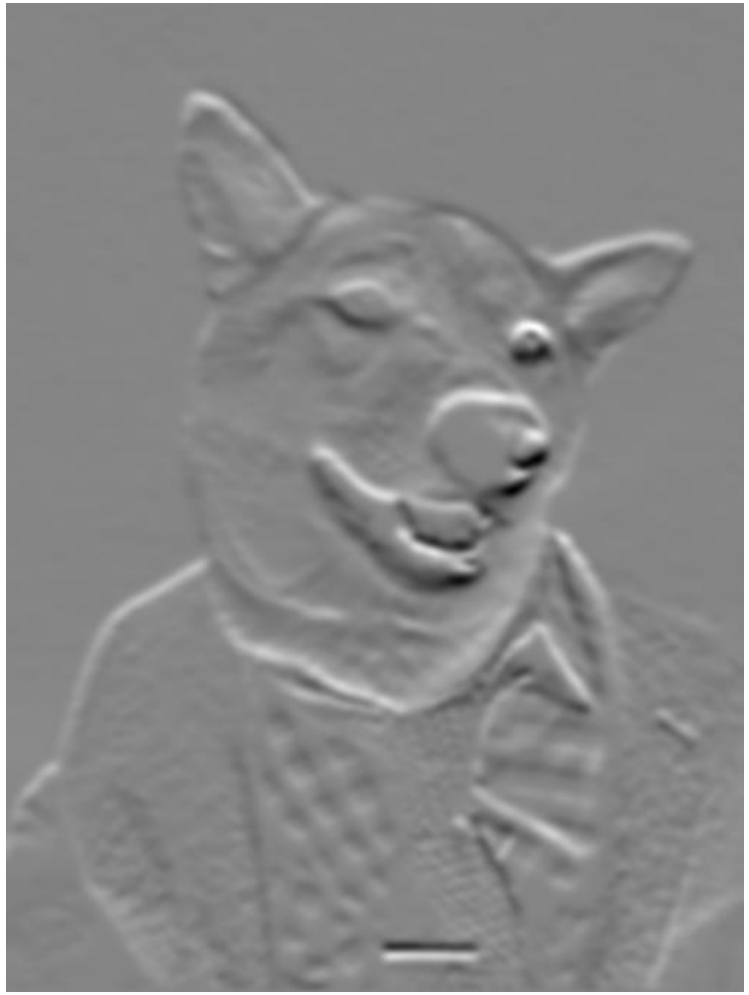
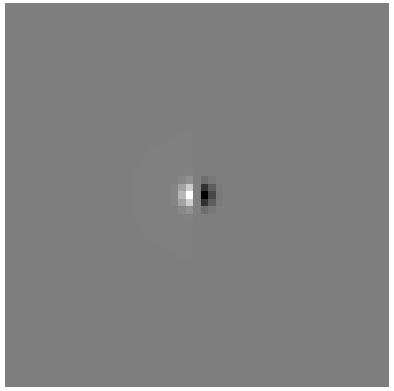


Image Convolution Examples

Gabor Filter
(Gaussian multiplied by sine or cosine)

*



=

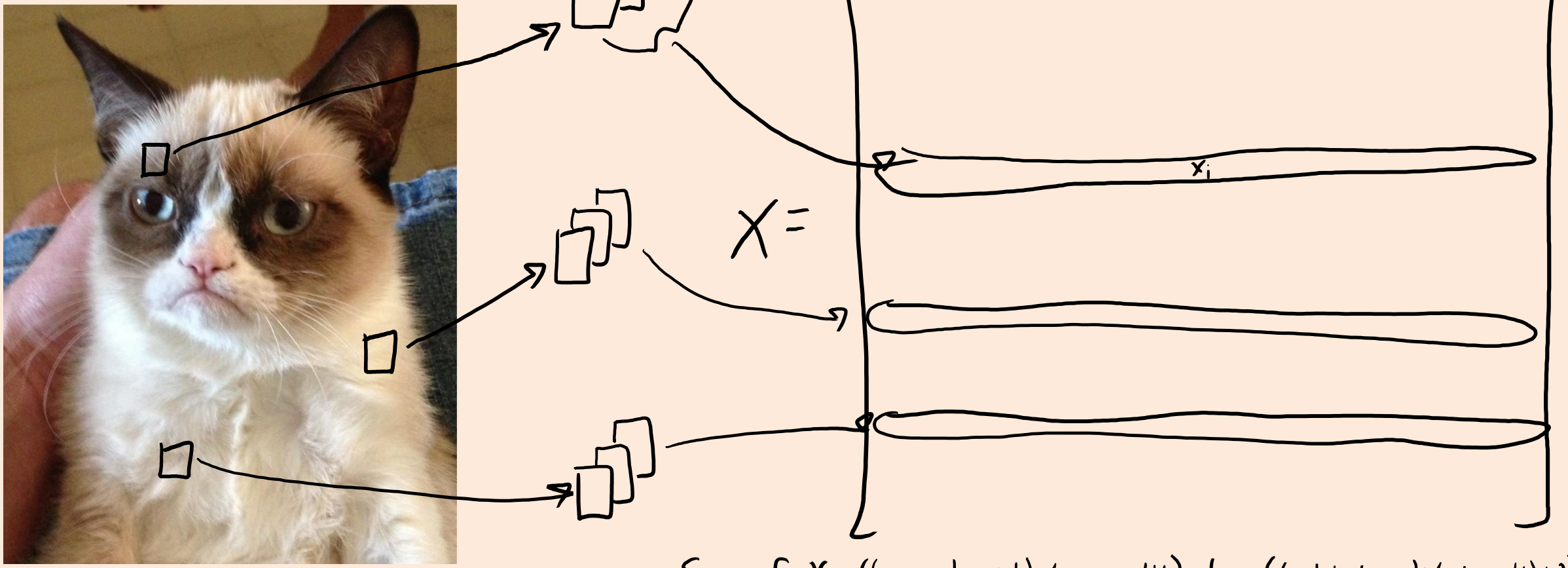
(smaller variance)

Vertical orientation

- Can obtain other orientations by rotating.
- May be similar to effect of V1 "simple cells."

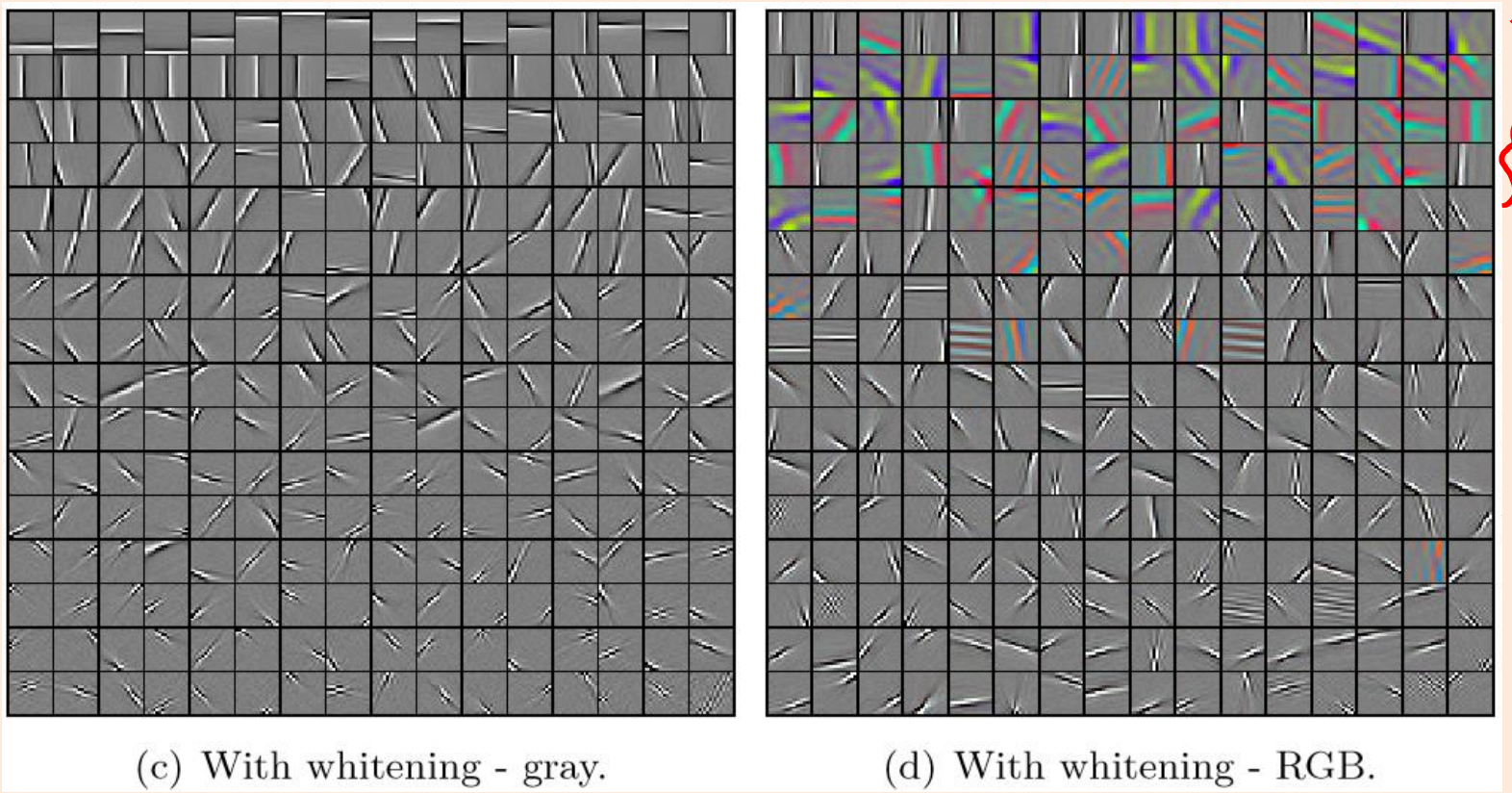
Unsupervised Learning of Filters for Image Patches

- Consider building an unsupervised model of image patches:



Unsupervised Learning of Filters for Image Patches

- Some methods to do this generate Gaussian/LoG/Gabor filters:
  - These filters are motivated from both neuroscience and ML experiments.

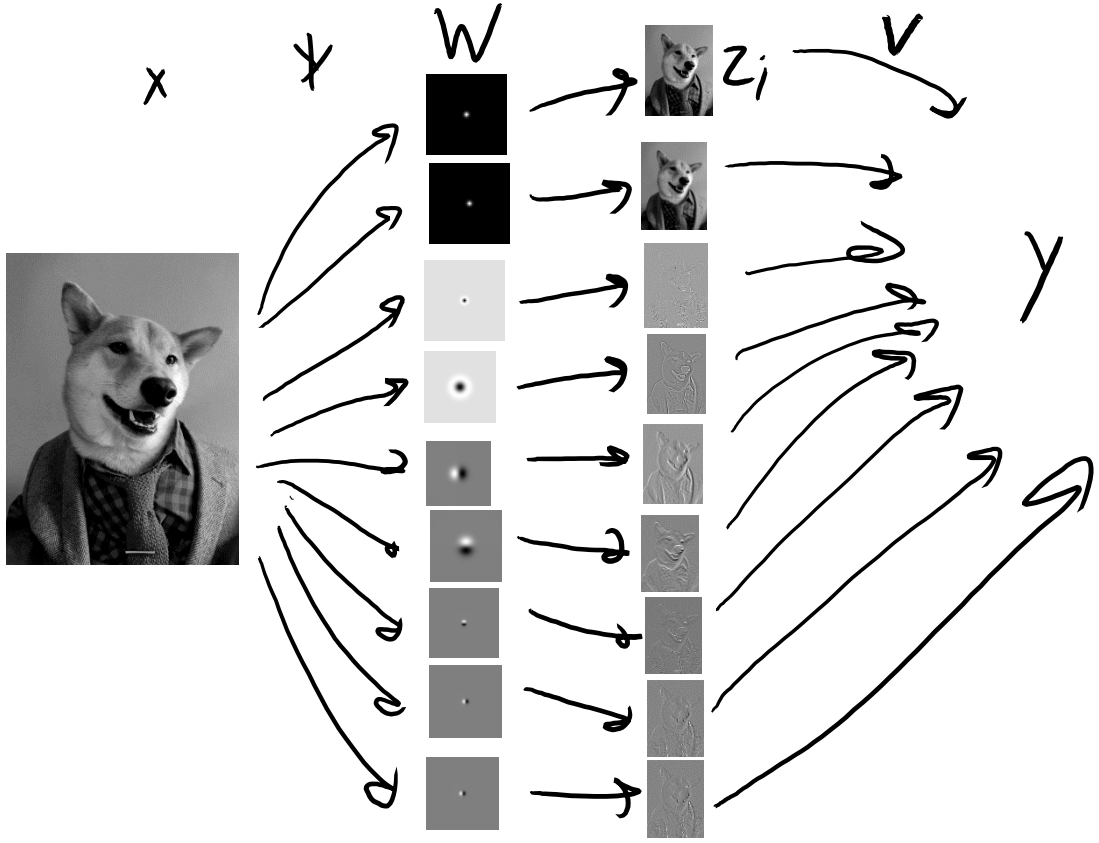


(c) With whitening - gray. (d) With whitening - RGB.

http://lear.inrialpes.fr/people/mairal/resources/pdf/review_sparse_arxiv.pdf

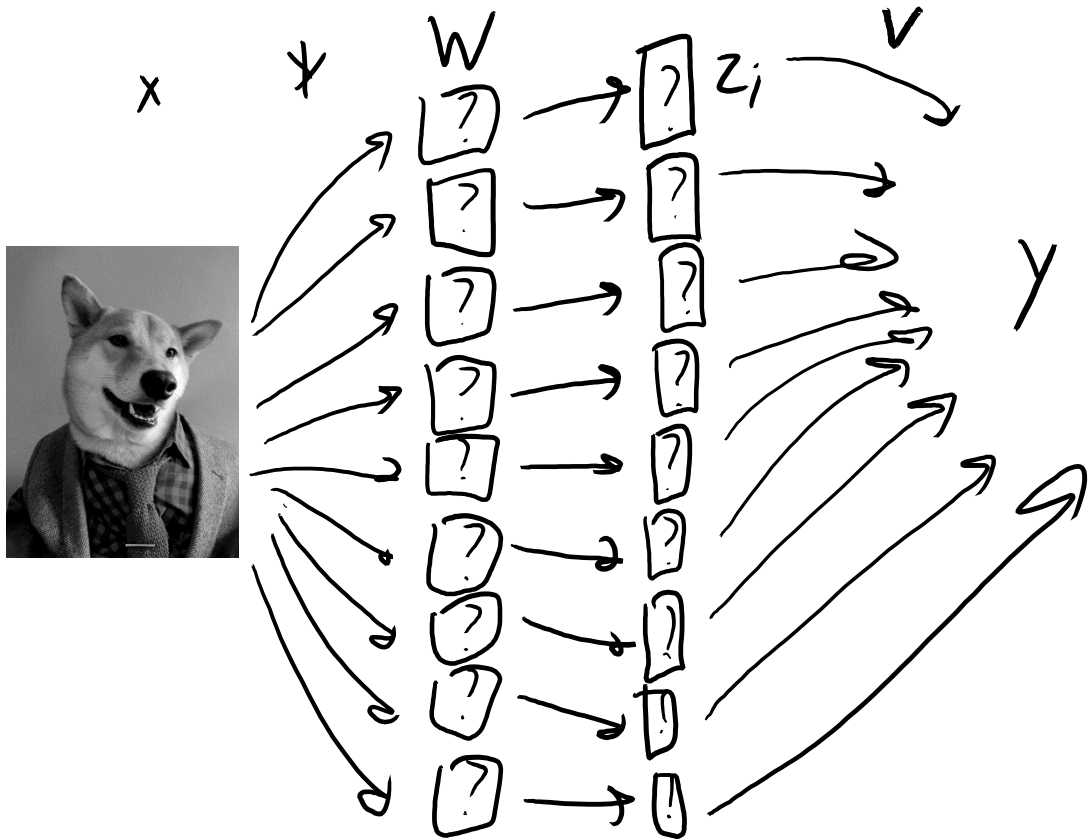
Motivation for Convolutional Neural Networks

- Classic vision methods uses fixed convolutions as features:
  - Usually have different types/variances/orientations.
  - Can do subsampling or take maxes across locations/orientations/scales.



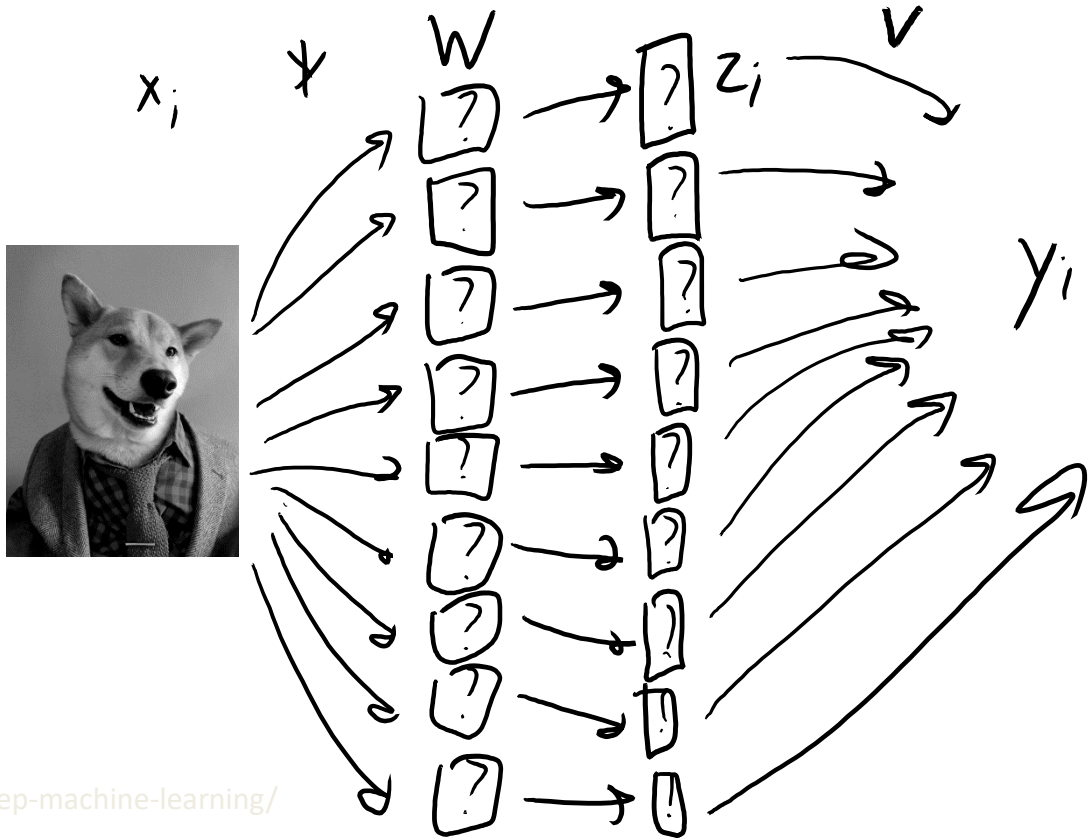
Motivation for Convolutional Neural Networks

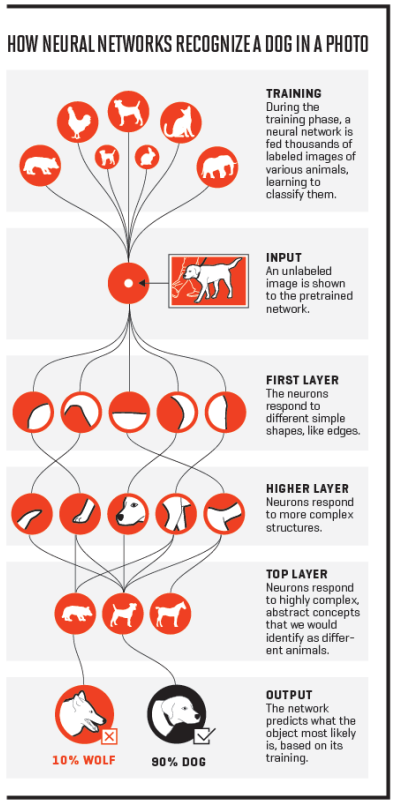
- Convolutional neural networks learn the convolutions:
  - Learning ‘W’ and ‘v’ automatically chooses types/variances/orientations.
  - Don’t pick from fixed convolutions, but learn the elements of the filters.



Motivation for Convolutional Neural Networks

- Convolutional neural networks learn the convolutions:
  - Learning ‘W’ and ‘v’ automatically chooses types/variances/orientations.
  - Can do multiple layers of convolution to get deep hierarchical features.

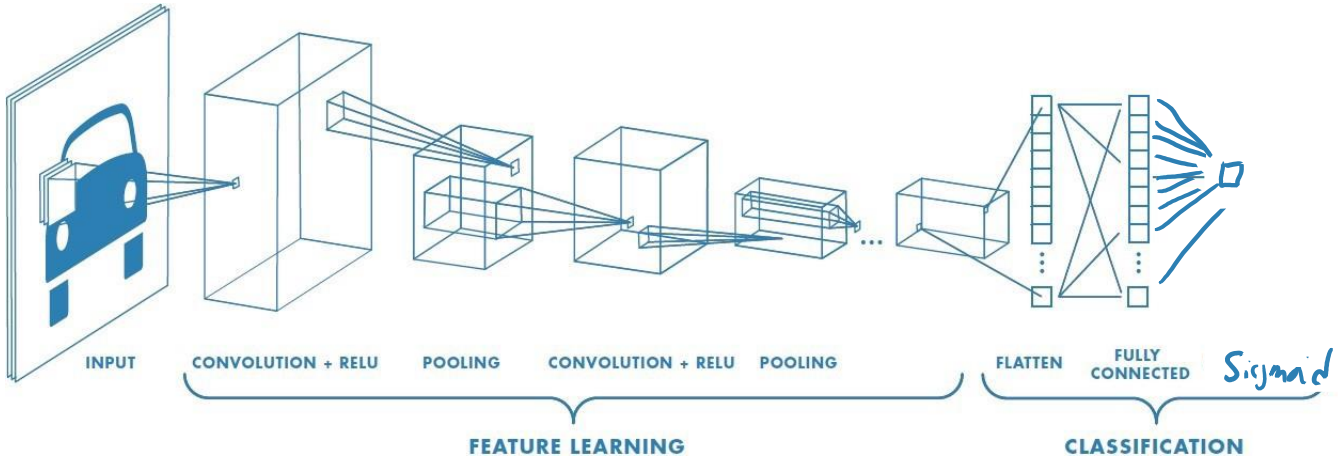




http://fortune.com/ai-artificial-intelligence-deep-machine-learning/

Convolutional Neural Networks

- Classic architecture of a convolutional neural network:

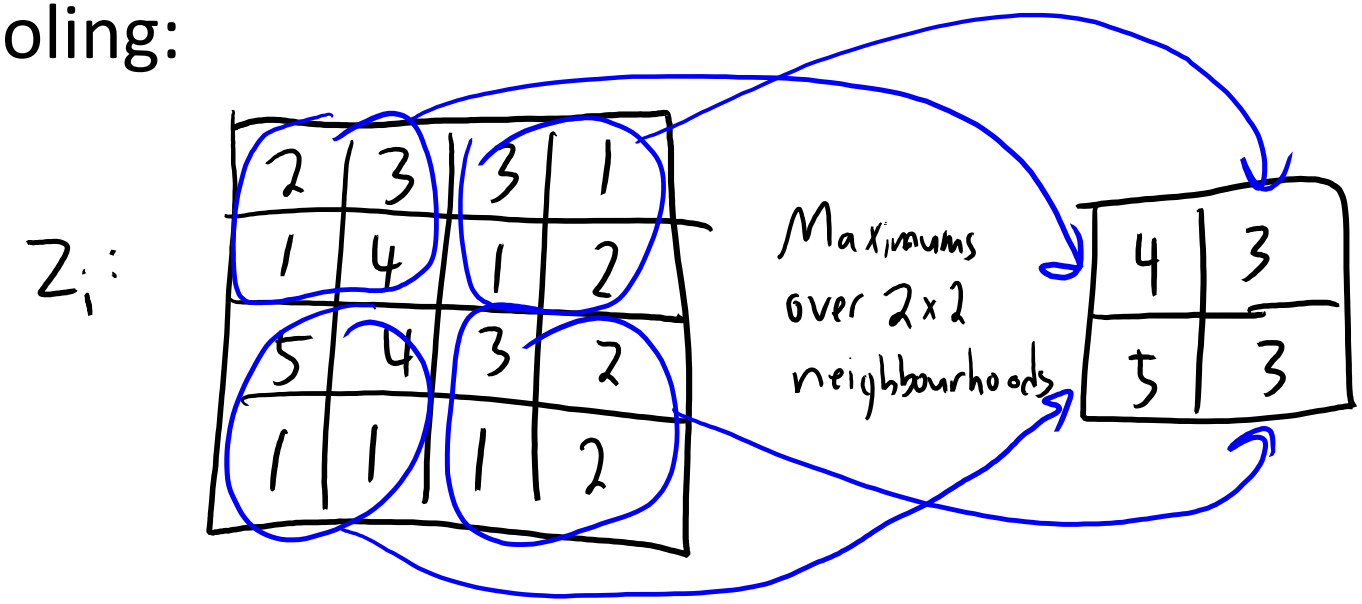


- Convolution layers:
  - Apply convolution with several different filters.
  - Sometimes these have a "stride": skip several pixels between applying filter.
- Pooling layers:
  - Aggregate regions to create smaller images (usually "max pooling").
- Fully-connected layers: usual "multiplication by W^l" in layer.

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Max Pooling Example

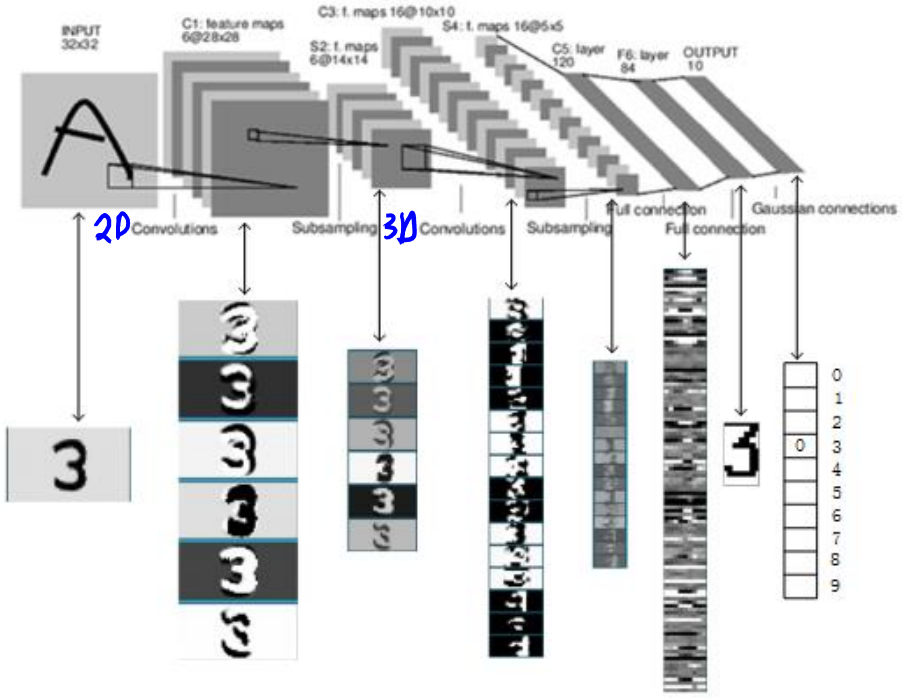
- Max pooling:



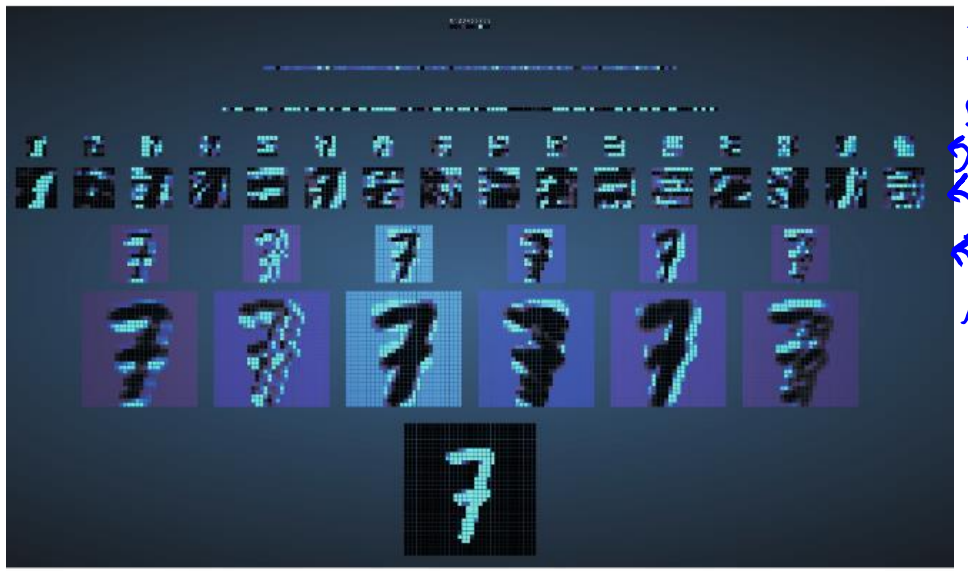
- Decreases size of hidden layer, so we need fewer parameters.
  - Gives some local translation invariance:
    - The precise location of max is not important.
- This is continuous and piecewise-linear but non-differentiable.
  - Like ReLU, we can still optimize this type of objective with SGD.

LeNet Convolutional Neural Networks

- Classic convolutional neural network (LeNet):



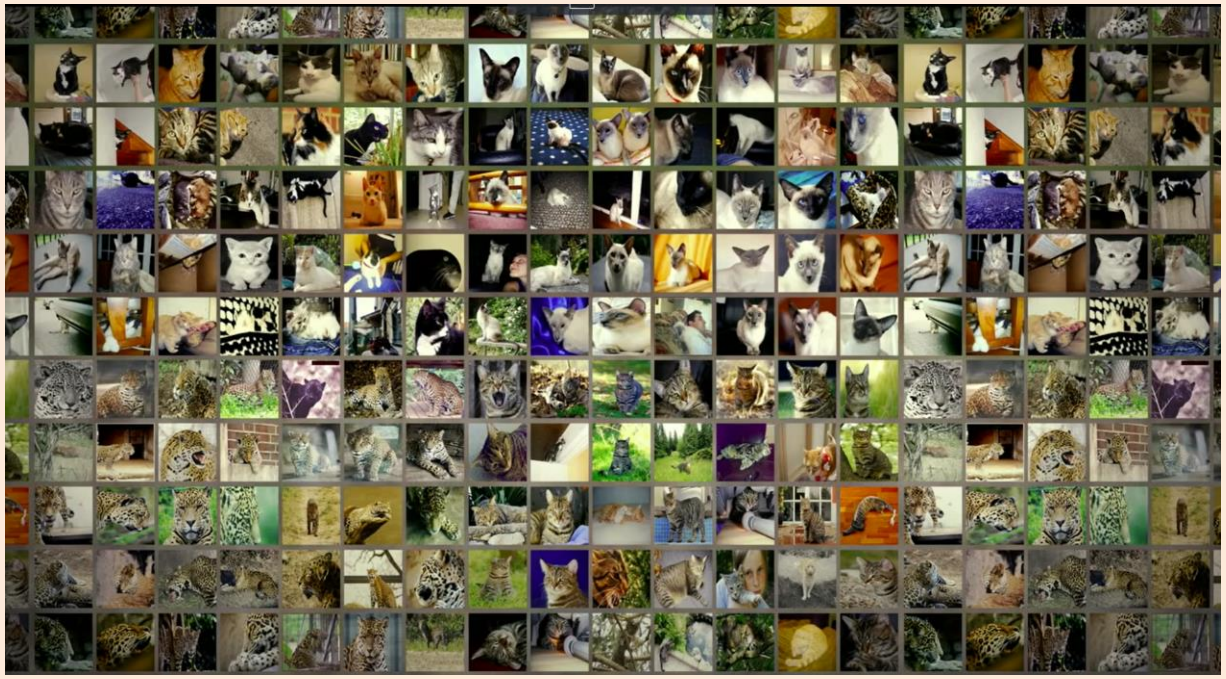
- Visualizing the “activations”:
  - http://scs.ryerson.ca/~aharley/vis/conv
  - http://cs231n.stanford.edu



http://scs.ryerson.ca/~aharley/vis/harley_vis_isvc15.pdf

ImageNet Competition

- ImageNet: Millions of labeled images, 1000 object classes.
  - Task is to classify images into one of the 1000 class labels.
    - We will discuss multi-class classification in Part 2 of the course.
  - Everyone submits their “best” model, winners announced.



https://www.youtube.com/watch?v=40riCqvRoMs

AlexNet Convolutional Neural Network

- Modern CNN era started with AlexNet (won 2012 competition):
  - 15.4% error vs. 26.2% for closest competitor.
  - 5 convolutional layers.
  - 3 fully-connected layers.
  - SG with momentum.
  - ReLU non-linear functions.
  - Data translation/reflection/ cropping.
  - L2-regularization + Dropout.
  - 5-6 days on two GPUs.

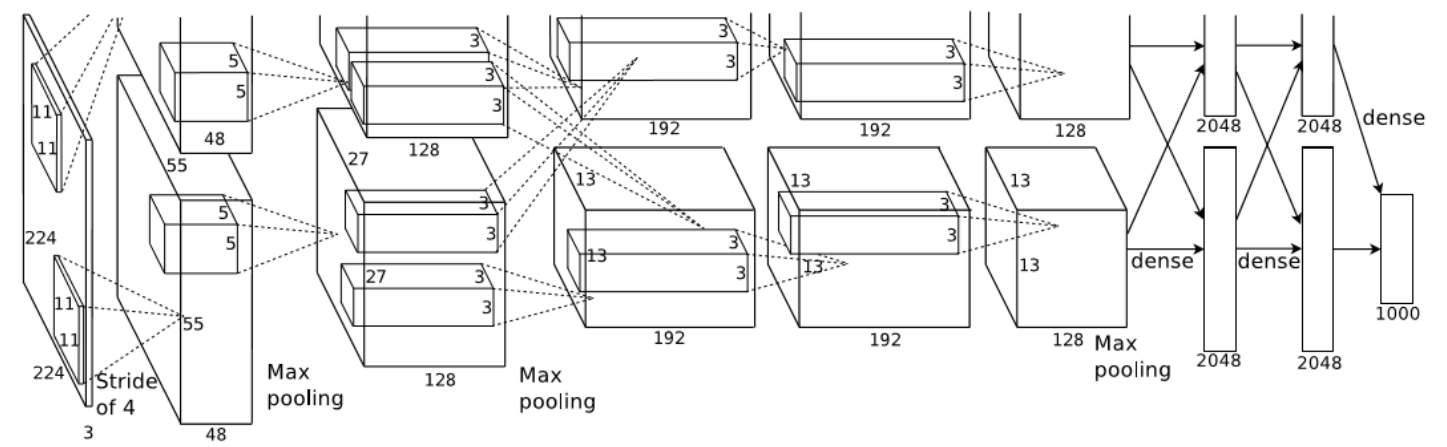
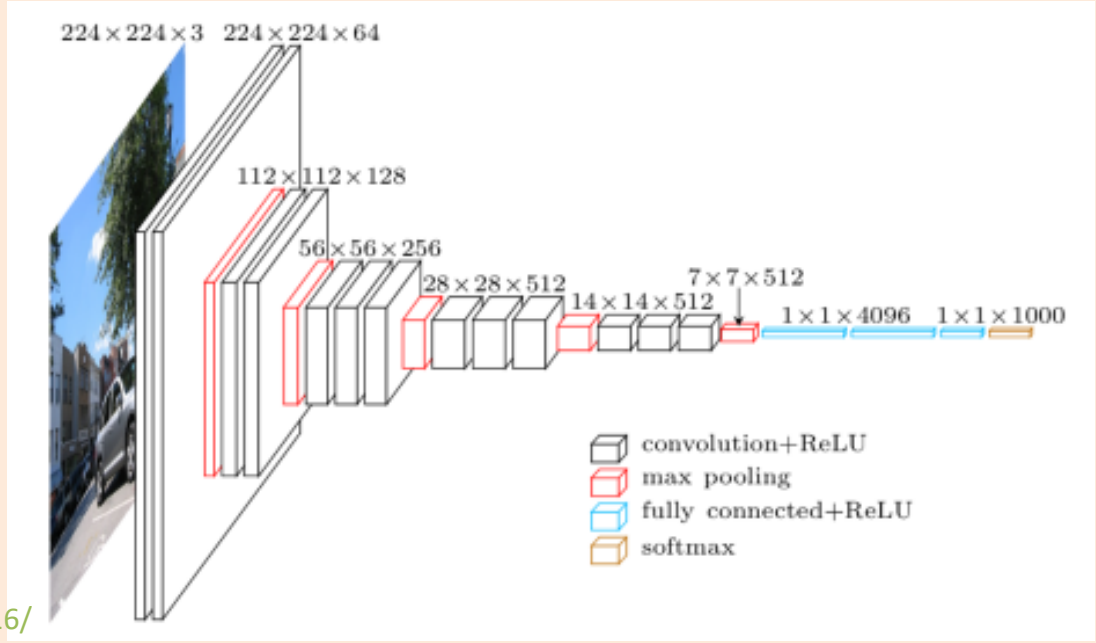


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

ImageNet Insights

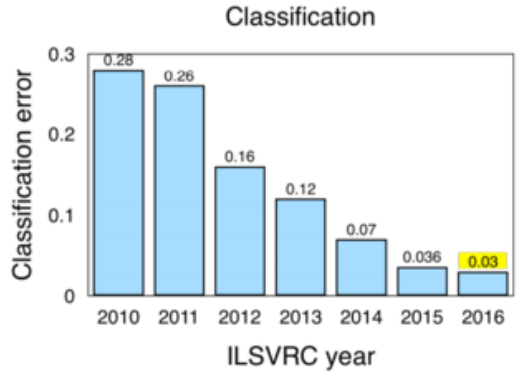
- Filters and stride got smaller over time.
  - Popular VGG approach uses 3x3 convolution layers with stride of 1.
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.
    - Also increases number of non-linear ReLU operations.



https://www.cs.toronto.edu/~frossard/post/vgg16/

ImageNet Insights

- Filters and stride got smaller over time.
  - Popular VGG approach uses 3x3 convolution layers with stride of 1.
  - GoogLeNet used multiple filter sizes ("inception layer"), but not as popular.
- Eventual switch to "fully-convolutional" networks.
  - No fully connected layers.
- ResNets allow easier training of deep networks.
  - Won all 5 tasks in 2015, training 152 layers for 2-3 weeks on 8 GPUs.
- Ensembles help.
  - 2016 winner combined predictions of previous networks.
- Competition ended in 2017!



http://www.themtank.org/a-year-in-computer-vision

Discussion of CNNs

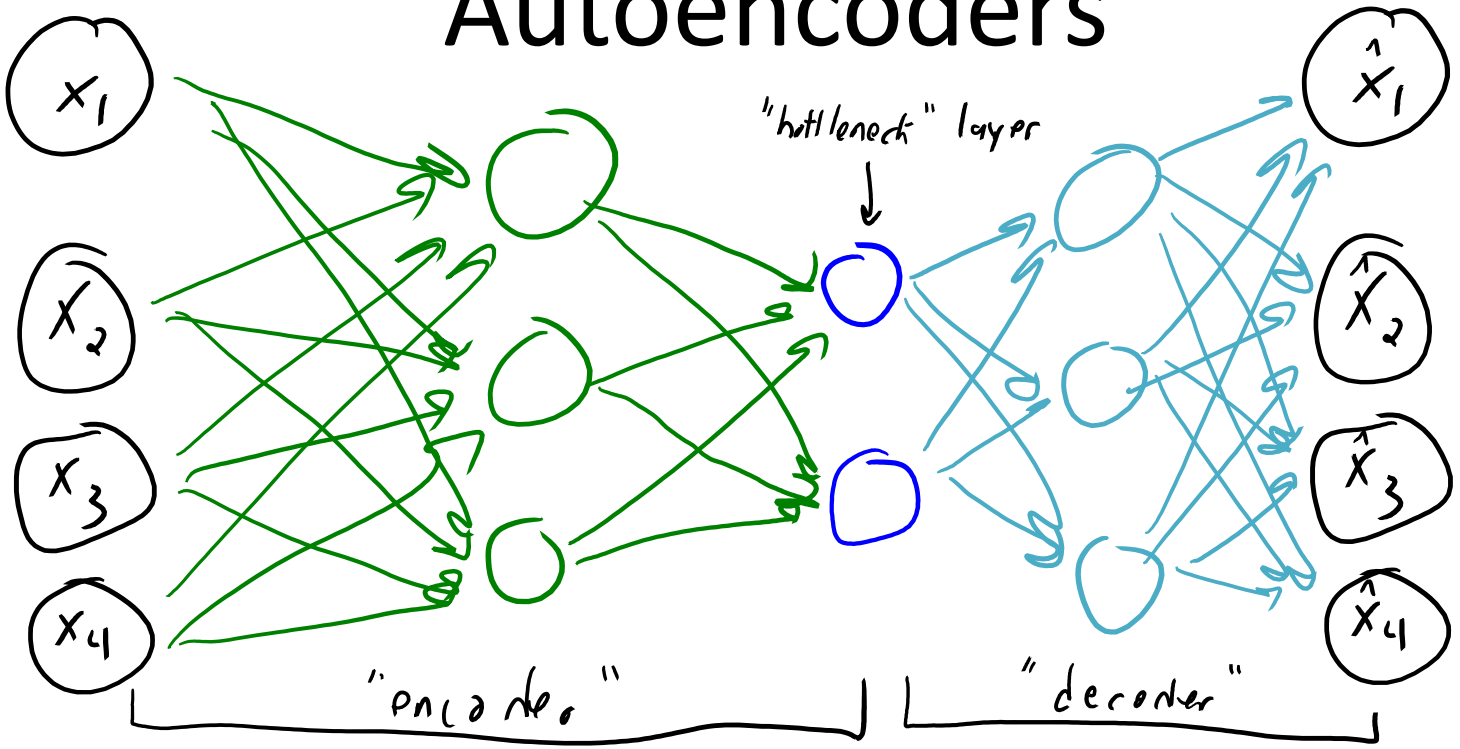
- Convolutional layers reduce the number of parameters in two different ways:
  - Each hidden unit only depends on small number of inputs from previous layer.
  - We use the same filters across the image.
    - So we do not learn a different weight for each "connection" like in classic neural networks.

- CNNs give some amount of translation invariance:
  - Because the filters are used across the image, they can detect a pattern anywhere in the image.
    - Even in image locations where the pattern has never been seen.
  - The pooling layer can also give some local invariance, against small translations of the image.

- CNNs are not only for images!
  - Can use CNNs for 1D sequences like sound or language.
  - Can use CNNs for 3D objects like videos or medical image volumes.
  - Can use CNNs for graphs.
- But you do need some notion of "neighbourhood" for convolutions to make sense.

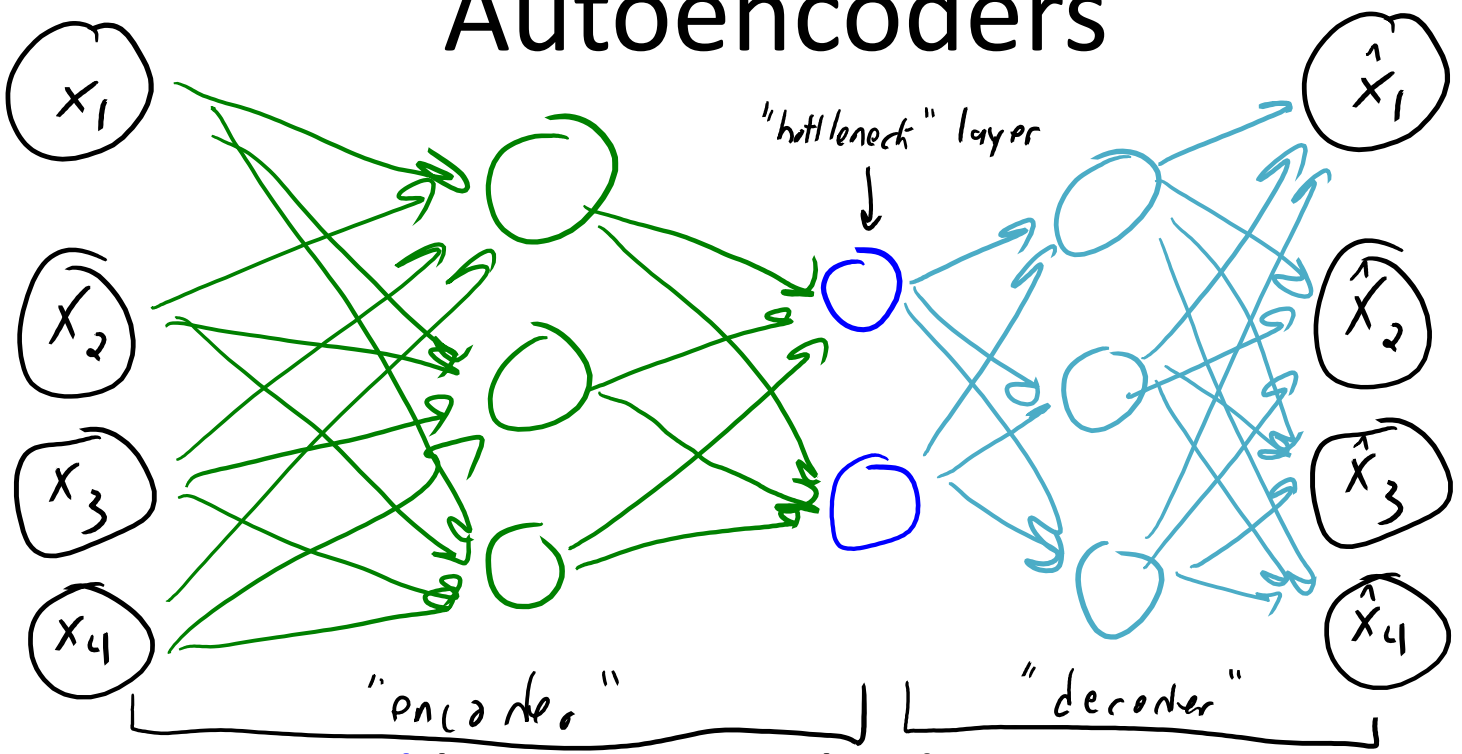
Next Topic: Autoencoders

Autoencoders



- Autoencoders are neural networks with same input and output.
  - Includes a bottleneck layer: with dimension ‘k’ smaller than input ‘d’.
  - First layers “encode” the input into bottleneck.
  - Last layers “decode” the bottleneck into a (hopefully valid) input.

Autoencoders



- This is an unsupervised learning method.
  - There are no labels 'y'.
- Relationship to principal component analysis (PCA):
  - With squared error and linear network, equivalent to PCA.
    - Size of bottleneck layer gives number of latent factors 'k' in PCA.
  - With non-linear transforms: a non-linear/deep generalization of PCA.

Summary

- Convolutions are flexible class of signal/image transformations.
  - Can approximate derivatives and integrals at different scales/orientations.
- Convolutional neural networks:
  - Include layers that apply several (learned) convolutions.
  - Significantly decreases number of parameters.
  - Achieves a degree of translation invariance.
  - Often combined with pooling operations like max pooling.
- Autoencoders:
  - Neural network where the output is the input.
  - Non-linear generalization of PCA.

- Next time: add colour to images.